

# Axel Polleres

*axel.polleres@wu.ac.at*

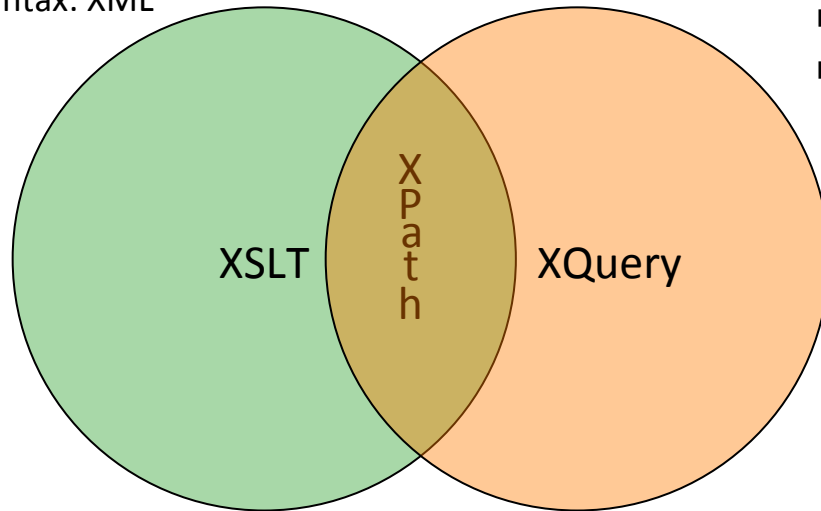
XQuery and SPARQL... Some  
thoughts...

# Connection points to tie together Query languages

- E.g., XQUERY+SPARQL = XSPARQL
- For those who haven't looked at it: SPARQL1.1
- Latest:
  - Other formats: RDB, CSV JSON
  - Streams, Windows and Big Data

# Transformation and Query Languages

XML Transformation  
Language  
Syntax: XML



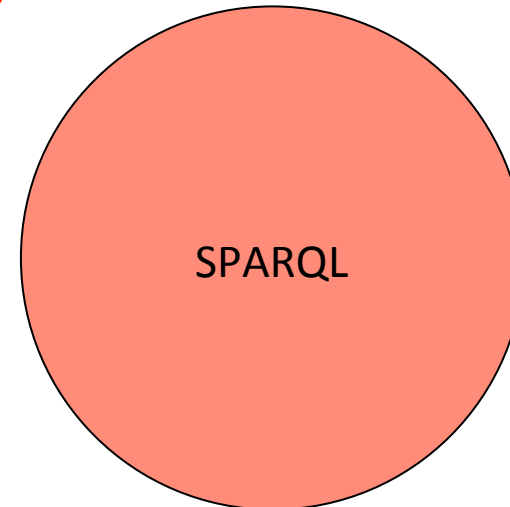
- XPath is the common core
- Mostly used to select nodes from an XML doc

- XML Query Language
- non-XML syntax

XML world

／ RDF world

- Query Language for RDF
- Pattern based
- declarative

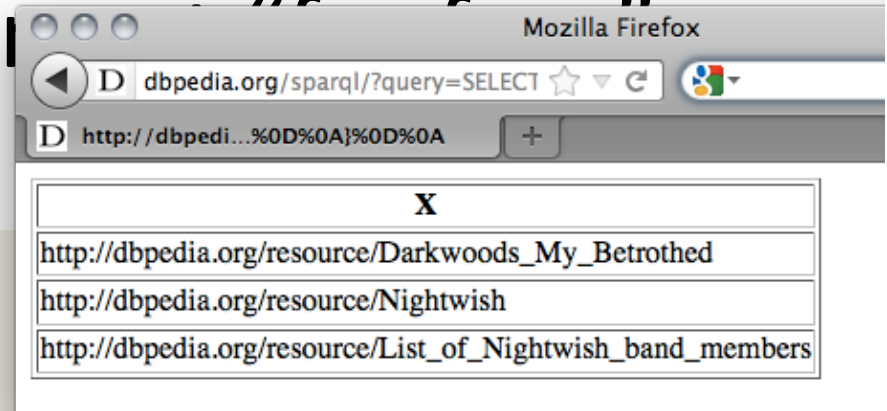


SPARQL XML Result format

RDF/XML... ambiguous

# SPARQL + Linked Data give you Semantic search and “fuzzy”

- *Which bands origin from Kitee?*



```
SELECT ?X
WHERE
{
  ?X <http://dbpedia.org/property/origin> <http://dbpedia.org/resource/Kitee>
}
```



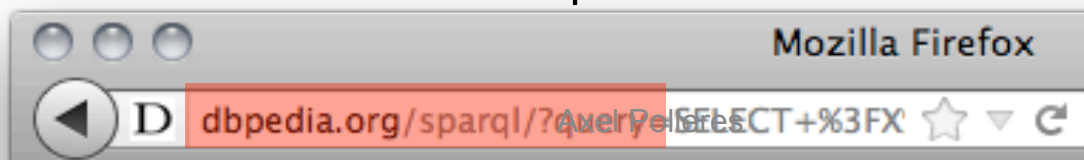
- *Try it out at <http://dbpedia.org/sparql>*

# SPARQL – Standard RDF Query Language and Protocol

- SPARQL (2008):

```
SELECT ?X
WHERE
{
  ?X <http://dbpedia.org/property/origin> <http://dbpedia.org/resource/Kitee>
}
```

- SQL “Look-and-feel” for the Web
- Essentially “graph matching” by *triple patterns*
- Allows conjunction (.), disjunction (UNION), optional (OPTIONAL) patterns and filters (FILTER)
- Construct new RDF from existing RDF
- Solution modifiers (DISTINCT, ORDER BY, LIMIT, ...)
- A **standardized** HTTP based protocol:



# Missing features in SPARQL1.0 (and why SPARQL1.1 was needed)

Based on implementation experience, in 2009 new W3C SPARQL WG founded to address common feature requirements requested urgently by the community:

[http://www.w3.org/2009/sparql/wiki/Main\\_Page](http://www.w3.org/2009/sparql/wiki/Main_Page)

1. Negation
2. Assignment/Project Expressions
3. Aggregate functions (SUM, AVG, MIN, MAX, COUNT, ...)
4. Subqueries
5. Property paths
6. Federated Queries
7. Updates
8. Entailment Regimes (RDFS, OWL,...)

- Other issues for wider usability:
  - Result formats (JSON, CSV, TSV),
  - Graph Store Protocol (REST operations on graph stores)
- ***SPARQL 1.1 W3C Recommendation since March 2013***

2012-04-17

# e.g., 6. Federated Queries in SPARQL1.1

*Find which persons in DBpedia have the same birthday as Axel (foaf-file):*

*SPARQL 1.1 has new feature SERVICE to query remote endpoints*

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
{ [ foaf:birthday ?MyB ].

  SERVICE <http://dbpedia.org/sparql> { SELECT ?N WHERE {
    [ dbpedia2:born ?B; foaf:name ?N ]. } }

  FILTER ( Regex(Str(?B),str(?MyB)) )
}
```

Doesn't work in practice, ☹ as SERVICE endpoints often only returns limited results...

# e.g., 6. Updates

- SQL has not only a query language, but also a Data manipulation language.  
→ SPARQL Update to fill this gap:

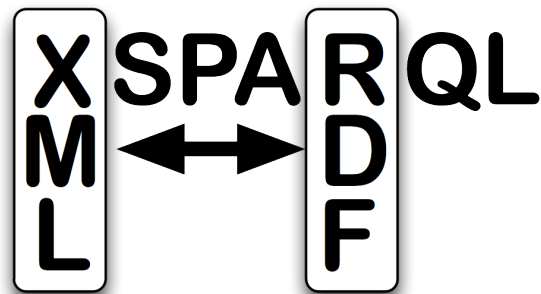
```
PREFIX ex: <http://example.org/>
DELETE { ?Item ex:price ?Pr }
INSERT { ?Item ex:price ?NewPr }
WHERE { ?Item ex:price ?Pr
        BIND (?Pr * 1.1 AS ?NewPr ) }
```

- Allows to change/update an RDF Store from outside, again via standard HTTP protocol.



# XSPARQL

**Idea:** One approach to conveniently query XML and RDF side-by-side: XSPARQL



- Transformation language
- Consume and generate XML and RDF
- Syntactic extension of XQuery, ie.

$XSPARQL = XQuery + SPARQL$



# XSPARQL Language Specification

W3C Member Submission 20 January 2009

**This version:**

<http://www.w3.org/Submission/2009/SUBM-xsparql-language-specification-20090120/>

**Latest version:**

<http://www.w3.org/Submission/xsparql-language-specification/>

**Authors:**

[Axel Polleres](#) - [DERI, NUI Galway](#)

[Thomas Krennwallner](#) - [Institute of Information Systems, Vienna University of Technology](#)

[Nuno Lopes](#) - [DERI, NUI Galway](#)

[Jacek Kopecký](#) - [University of Innsbruck](#)

[Stefan Decker](#) - [DERI, NUI Galway](#)

Copyright © 2009 by [DERI Galway](#) at the [National University of Ireland, Galway](#), Ireland.

This work is supported by [Science Foundation Ireland](#) under grants number SFI/02/CE1/I131 and SFI/08/CE/I1380 and under the European Commission European FP6 project [inContext](#) (IST-034718).

This document is available under the [W3C Document License](#). See the [W3C Intellectual Rights Notice and Legal Disclaimers](#) for additional information.



# XSPARQL: Syntax overview (I)

Prefix declarations

<b>P</b>	declare namespace <i>prefix</i> ="namespace-URI" or prefix <i>prefix</i> : <namespace-URI>
----------	---

Body:

<b>F</b>	for <i>var</i> [at <i>posVar</i> ] in <i>FLOWR'</i> expression
<b>L</b>	let <i>var</i> := <i>FLWOR'</i> expression
<b>W</b>	where <i>FLWOR'</i> expression
<b>O</b>	order by <i>FLWOR'</i> expression

Data Input  
(XML or RDF)

<b>F'</b>	for <i>varlist</i> [at <i>posVar</i> ]
<b>D</b>	from / from named ( <dataset-URI> or <i>FLWOR'</i> expr.)
<b>W</b>	where { <i>pattern</i> }
<b>M</b>	order by <i>expression</i> limit <i>integer</i> > 0 offset <i>integer</i> > 0

or

Data Output  
(XML or RDF)

<b>C</b>	construct { <i>template</i> (with nested <i>FLWOR'</i> expressions) }
<b>R</b>	return <i>XML</i> + nested <i>FLWOR'</i> expressions

or

# XSPARQL Syntax overview (II)

XQuery or SPARQL prefix declarations  
Any XQuery query

SPARQLFOR Clause represents a SPARQL query

construct allows to create RDF

<b>P</b>	declare namespace <i>prefix</i> ="namespace-URI" or prefix <i>prefix</i> : <namespace-URI>	
<b>F</b> <b>L</b> <b>W</b> <b>O</b>	for var [at <i>posVar</i> ] in <i>FLOWR'</i> expression let var := <i>FLWOR'</i> expression where <i>FLWOR'</i> expression order by <i>FLWOR'</i> expression	<b>or</b>
<b>F'</b> <b>D</b> <b>W</b> <b>M</b>	for varlist [at <i>posVar</i> ] from / from named ( <dataset-URI> or <i>FLWOR'</i> expr.) where { <i>pattern</i> } order by <i>expression</i> limit <i>integer</i> > 0 offset <i>integer</i> > 0	<b>or</b>
<b>C</b>	construct { <i>template (with nested FLWOR' expressions)</i> }	<b>or</b>
<b>R</b>	return XML+ nested <i>FLWOR'</i> expressions	

# Federated Queries in SPARQL1.1

*Find which persons in DBPedia have the same birthday as Axel (foaf-file):*

*SPARQL 1.1 has new feature SERVICE to query remote endpoints*

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
{ [ foaf:birthday ?MyB ].

SERVICE <http://dbpedia.org/sparql> { SELECT ?N WHERE {
  [ dbpedia2:born ?B; foaf:name ?N ]. FILTER ( Regex(str(?B),str(?MyB)) ) } }
}
```

**Doesn't work!!! ?MyB unbound in SERVICE query**

# e.g. Federated Queries

*Find which persons in DBPedia have the same birthday as Axel (foaf-file):*

## *In XSPARQL:*

```
prefix dbprop: <http://dbpedia.org/property/>  
prefix foaf: <http://xmlns.com/foaf/0.1/>  
prefix : <http://xsparql.deriv.org/bday#>
```

```
let $MyB := for * from <http://polleres.net/foaf.rdf>  
  where { [ foaf:birthday $B ]. }  
  return $B
```

```
for * from <http://dbpedia.org/> endpoint <http://dbpedia.org/sparql>  
where { [ dbprop:born $B; foaf:name $N ].  
  filter ( regex(str($B),str($MyB)) ) }  
construct { :axel :sameBirthDayAs $N }
```

Specifies the endpoint to perform the query, similar to SERVICE in SPARQL1.1

Works! In XSPARQL bound values (?MyDB) are **injected** into the SPARQL subquery  
→ More direct control over “query execution plan”

# Connection points to tie together languages&formats

- Other formats: RDB, CSV JSON
  - RDB2RDF, REC out since Sept 2012  
<http://www.w3.org/TR/r2rml/>  
mapping language to transform RDB to RDF
  - JSON-LD, REC out since Jan 2014  
<http://www.w3.org/TR/json-ld/>
  - CSV on the Web WG  
<http://www.w3.org/2013/05/lcsv-charter.html>
- Streams, Windows and Big Data:
  - <http://www.w3.org/community/rsp/> RDF Stream Processing Community Group
  - Related?  
<http://www.w3.org/TR/xquery-30/#id-windows>
- Binary format/compression is becoming an issue:
  - EXI vs. HDT. <http://www.w3.org/Submission/2011/SUBM-HDT-20110330/>



Binary RDF Representation for Publication and Exchange (HDT)

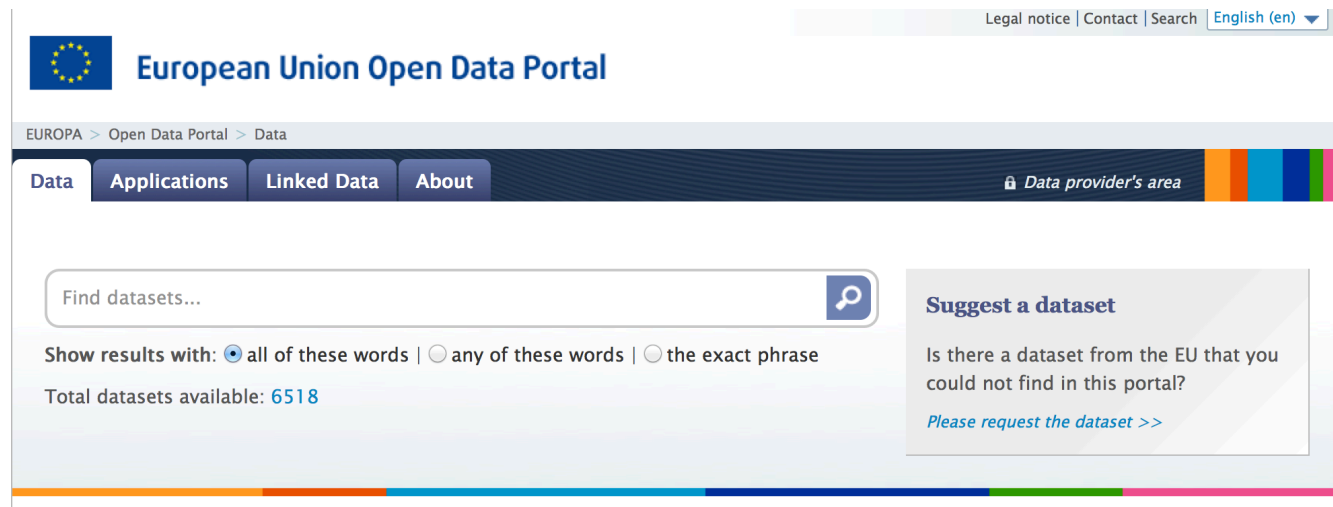
W3C Member Submission 30 March 2011

This version:

# Linked Data, RDF and SPARQL...

## Why at all?

- Standard protocol and standard means to integrate different datasets
- Standard way to query the data and the schema alongside (Schema & Data decoupled)
- „share [...] data in ways that are easily discoverable, useable, or understandable by the public“
- Increasingly supported/used in Open Data, e.g.



The screenshot shows the European Union Open Data Portal search interface. At the top, there is a navigation bar with the European Union flag and the text "European Union Open Data Portal". Below this, a breadcrumb trail reads "EUROPA > Open Data Portal > Data". A dark blue navigation bar contains tabs for "Data", "Applications", "Linked Data", and "About", along with a "Data provider's area" link. The main content area features a search bar with the placeholder text "Find datasets...". Below the search bar, there are radio buttons for "Show results with:" and the option "all of these words" is selected. The text "Total datasets available: 6518" is displayed. On the right side, there is a "Suggest a dataset" section with the text "Is there a dataset from the EU that you could not find in this portal?" and a link "Please request the dataset >>".